

Python 101 x SciPy

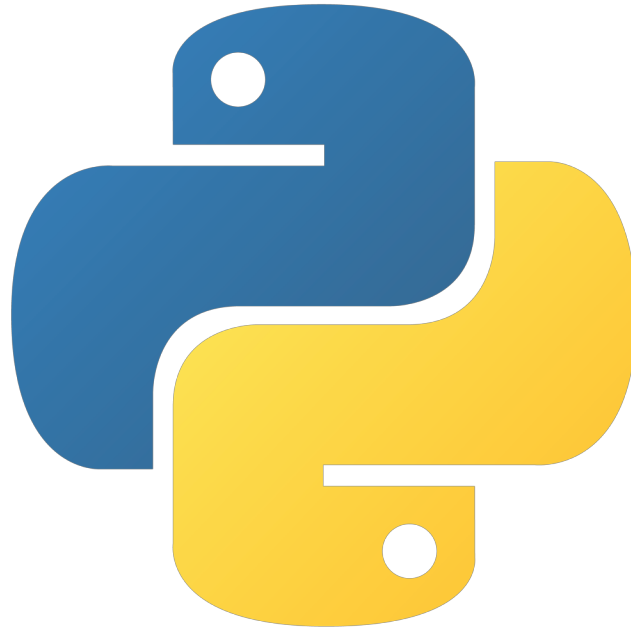
(5) Meet SciPy

AIAT Academy

Based on stanford.edu/~schmit/cme193

SciPy's Outline

- NumPy
- SciPy
- Matplotlib



Why Python and SciPy?

- Python is a **widely used, general purpose** programming language
- **Easy to start** working with
- **Scientific computation similar** to MATLAB and Octave
- Used by major **Deep Learning frameworks** such as TensorFlow and PyTorch

NumPy

NumPy

- Fundamental package for scientific computing with Python
- N-dimensional array object
- Linear Algebra, Fourier transform, etc.
- Open source

- Optimized library for matrix and vector computation
(Lots of computation can be efficiently represented as vectors)

Main data type: **`np.ndarray`**

The data type for representing matrix/vector computation

Constructor function: **`np.array()`**

NumPy Basics (np.array)

```
import numpy as np
```

```
A = np.array([[1, 2, 3], [4, 5, 6]])
```

```
# >> A == array([ [1 2 3],
#                [4 5 6] ] )
```

```
A_float = np.array([1, 2 ,3], float)
```

NumPy Basics (np.arange)

```
import numpy as np
```

```
np.arange(3)           # >> array([0, 1, 2])
```

```
np.arange(3.0)         # >> array([0., 1., 2.])
```

```
np.arange(3, 7)        # >> array([3, 4, 5, 6])
```

```
np.arange(3, 7, 2)     # >> array([3, 5])
```

```
numpy.arange(start, stop, step)
```


NumPy Basics (np.zeros)

```
import numpy as np
```

```
A = np.zeros((2, 3))
```

```
# >> array( [ [0., 0., 0.],  
              [0., 0., 0.] ] )
```

```
A.shape
```

```
# >> (2, 3)
```

numpy.zeros(shape)

NumPy Basics (np.zeros)

```
np.random.random((2, 3))
```

```
# array([[ 0.78084261, 0.64328818, 0.55380341],
         [ 0.24611092, 0.37011213, 0.83313416]])
```

```
np.random.normal(loc=1.0, scale=2.0, size=(2, 2))
```

```
# array([[ 2.87799514, 0.6284259 ],
         [ 3.10683164, 2.05324587]])
```

NumPy Basics (Array Attributes)

```
a = np.arange(10).reshape((2, 5))
```

<code>a.ndim</code>	# 2 dimensions
<code>a.shape</code>	# (2, 5) shape of array
<code>a.size</code>	# 10 # of elements
<code>a.T</code>	# Transpose
<code>a.dtype</code>	# data type

NumPy Basics (Operations)

```
a = np.arange(4)
```

```
# >> array( [0, 1, 2, 3] )
```

```
b = np.array([2, 3, 2, 4])
```

```
a * b          # array( [ 0, 3, 4, 12] )
```

```
a - b          # array( [ 2, 2, 0, 1] )
```

```
c = [2, 3, 4, 5]
```

```
a * c          # array( [ 0, 3, 8, 15] )
```

NumPy Basics (Vector Operations)

- Inner product
- Outer product
- Dot product (Matrix multiplication)

Note: NumPy automatically converts lists

```
u = [1, 2, 3]
```

```
v = [1, 1, 1]
```

```
np.inner(u, v)          # 6
```

```
np.outer(u, v)
```

```
# array([[1, 1, 1],
```

```
#         [2, 2, 2],
```

```
#         [3, 3, 3]])
```

```
np.dot(u, v)           # 6
```

NumPy Basics (Vector Operations)

```
A = np.ones((3, 2)).T  
# array([[ 1.,  1.],  
#        [ 1.,  1.]])
```

```
B = np.ones((2, 3))  
# array([[ 1.,  1.,  1.],  
#        [ 1.,  1.,  1.]])
```

```
np.dot(A, B)
```

```
# array([[ 2.,  2.,  2.],  
#        [ 2.,  2.,  2.]])
```

```
np.dot(B, A)
```

```
# array([[ 3.,  3.],  
#        [ 3.,  3.]])
```

```
np.dot(B.T, A.T)
```

```
# array([[ 2.,  2.,  2.],  
#        [ 2.,  2.,  2.]])
```

SciPy

What is SciPy

SciPy is a library of algorithms and mathematical tools built to work with NumPy arrays.

- linear algebra - `scipy.linalg`
- statistics - `scipy.stats`
- optimization - `scipy.optimize`
- sparse matrices - `scipy.sparse`
- signal processing - `scipy.signal`
- etc.

SciPy (Linear Algebra)

- Slightly different from `numpy.linalg`. Always uses BLAS/LAPACK support, so could be faster.
- Some more functions.
- Functions can be slightly different.

SciPy (Optimization)

- General purpose minimization: CG, BFGS, least-squares
- Constrained minimization; non-negative least-squares
- Minimize using simulated annealing
- Scalar function minimization
- Root finding
- Check gradient function
- Line search

SciPy (Statistics)

- Mean, median, mode, variance, kurtosis
- Pearson correlation coefficient
- Hypothesis tests (ttest, Wilcoxon signed-rank test, Kolmogorov-Smirnov)
- Gaussian kernel density estimation

SciPy (Sparse)

- Sparse matrix classes: CSC, CSR, etc.
- Functions to build sparse matrices
- `sparse.linalg` module for sparse linear algebra
- `sparse.csgraph` for sparse graph routines

SciPy (Signal)

- Convolutions
- B-splines
- Filtering
- Continuous-time linear system
- Wavelets
- Peak finding

SciPy (I/O)

Methods for loading and saving data

- MATLAB files
- Matrix Market files (sparse matrices)
- Wav files

Matplotlib

What is Matplotlib

- Plotting library for Python
- Works well with NumPy
- Syntax similar to MATLAB

matplotlib

The logo for Matplotlib, which is a circular plot with a grid. Inside the plot, there are several colored segments (orange, yellow, green, blue) and a small purple line, resembling a stylized rose or a polar plot.

Examples of Plotting

